IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
WACO DIVISION

PASAFESHARE LLC, **Plaintiff,** **vs.** **MICROSOFT CORPORATION,** **Defendant.**	**No. 6:20-cv-00397-ADA**

DECLARATION OF WILLIAM ROSENBLATT IN SUPPORT OF DEFENDANT MICROSOFT CORPORATION'S RESPONSIVE CLAIM CONSTRUCTION BRIEF

TABLE OF CONTENTS

Page

At the request of the law firm of Perkins Coie LLP and on behalf of Defendant Microsoft Corporation ("Microsoft"), I hereby submit this Declaration. I am over 18 years of age and competent to testify to the matters set forth herein. The statements contained in this Declaration are based on my personal knowledge.

I. INTRODUCTION

1. My name is William Rosenblatt. I have been retained to assist the Court and provide expert testimony concerning the construction of certain claim terms in U.S. Patents Nos. 9,455,961 (the "'961 Patent"); 9,615,116 (the "'116 Patent"); and 10,095,848 (the "'848 Patent") (collectively the "Patents-in-Suit"). This Declaration contains a brief introduction to the relevant technology field as well as my opinions concerning some of the claim terms that I understand to be at issue in the above-captioned litigation.

2. This declaration contains statements of my opinions formed to date and the bases and reasons for those opinions. I may offer additional opinions based on further review of materials in this case, including opinions and/or testimony of other expert witnesses.

3. As part of my analysis, I reviewed the documents and materials identified in this Declaration, including the Patents-in-Suit, their prosecution histories, Plaintiff's Opening Claim Construction Brief, any extrinsic evidence cited, and any information or references discussed and/or identified in this Declaration. I have considered information from various other sources in forming my opinions. I have also drawn on my more than 25 years of experience in the area of digital rights management ("DRM").

4. This Declaration is based on the information currently available to me. To the extent that additional information becomes available, I reserve the right to continue my

investigation and study, which may include a review of documents and information that may be produced, as well as testimony from depositions that may not yet be taken.

5. Microsoft is compensating me for my time at $600 per hour or $900/hour for testimony by deposition or at trial. None of my compensation depends in any way on the outcome of this Litigation.

II. SUMMARY OF OPINIONS

6. The phrase "software instructions . . . executed by a processor" as used in all independent claims of the Patents-in-Suit should be construed according to its plain and ordinary meaning, which is "executable software instructions."

7. The term "Publisher Key (PK)" as used in all independent claims of the Patents-in-Suit is indefinite because a person of ordinary skill in the art would be unable to understand the scope of the term with reasonable certainty.

8. The phrase "enabling an authorized user communicating via a secure tunnel or virtual private network to modify PDP content via an editing mode associated with said LCV" as used in claims 6 and 14 of the '961 patent and claims 15 and 18 of the '848 patent should be construed to mean "if an authorized user is communicating via a secure tunnel or virtual private network, then enabling that authorized user to modify PDP content via an editing mode of the LCV."

9. The terms "enhanced user authentication or authorization (EUAA) information," "EUAA information," and "EUAA data" as used in claims 2–4, 12, and 13 of the '961 patent are indefinite because a person of ordinary skill in the art would be unable to understand the scope of these terms with reasonable certainty.

III. EXPERT QUALIFICATION AND PREVIOUS TESTIMONY

10. My CV is attached hereto as Exhibit A. It includes a list of all my prior expert testimony from the last four years, and a list of my prior publications in the last ten years.

11. My educational background includes a Bachelor of Science in Engineering degree in Electrical Engineering and Computer Science from Princeton University in 1983, a Master of Science of Computer and Information Science from the University of Massachusetts in 1990, and coursework toward a PhD in the latter subject.

12. My professional background includes over 25 years of experience in the field of digital rights management ("DRM"), which I understand to be the technical field of the Patents-in-Suit. I am the lead author of the book *Digital Rights Management: Business and Technology* (Wiley, 2001) as well as several journal articles, whitepapers, and other writings on DRM, including the chapter "Digital Rights and Digital Television" in the book Television Goes Digital (Springer, 2009).

13. I have consulted to many clients on DRM-related topics, including several vendors of DRM technology as well as major film studios, record labels, publishers, online service providers, and technology companies such as Adobe, Google, and Sony. I have helped design two DRM systems and contributed to various standards initiatives related to DRM.

14. In addition, I have consulted to or testified before public policy bodies including the U.S. Copyright Office, National Academies, Federal Trade Commission, European Commission, and French Ministry of Culture, and I have lectured on DRM-related subjects at universities and law schools including MIT, Columbia, University of Virginia, and the law schools of Rutgers and the University of New Hampshire. I have chaired conferences on technology issues related to copyright in the digital age since 2004 and have spoken on the

subject at other conferences including the World Economic Forum in Davos, Switzerland, and other events on four continents.

IV. LEGAL STANDARDS

15. I am not an attorney, but I have been informed about certain aspects of the law that I understand are relevant to my analysis and opinions set forth herein.

16. I understand that a patent's specification must conclude with one or more claims particularly pointing out and distinctly claiming the subject matter that the applicant regards as his invention. I also understand that patent claims are interpreted from the perspective of a person of ordinary skill in the art at the time of the invention ("POSA") in light of the intrinsic evidence. The intrinsic evidence includes the language of the claims themselves, the specification of the patent, and the relevant prosecution history from the United States Patent and Trademark Office ("USPTO"). Other evidence (such as dictionaries) not in the written record of the patent, and other extrinsic evidence also may be considered if it does not contradict the intrinsic evidence, but it is not necessary to rely on extrinsic evidence if the meaning of the claims is clear from the intrinsic evidence.

17. I understand that as a general matter a claim should not be limited to a preferred embodiment described in the specification. I also understand that a claim need not be interpreted to encompass all disclosed embodiments when the claim language is clearly directed to a subset of embodiments. I understand that a special definition for a claim term (differing from its plain and ordinary meaning to a person of skill in the art) may be clearly set forth in the specification or prosecution history and that the inventor's lexicography will govern in those instances.

18. I understand that claims generally may not be construed one way in order to obtain their allowance and in a different way to determine infringement. I also understand that the prosecution history of a patent and related patents can inform the meaning of a claim term by

demonstrating how the inventor and patent examiner interpreted the terms. For example, an applicant's repeated and consistent remarks during prosecution can shed light on the meaning of a claim term. I further understand that an applicant can make clear and unambiguous statements of disavowal or disclaimer of claim scope during prosecution such that the disavowed material is no longer within the scope of the claims. I understand that mere criticism of a particular embodiment otherwise covered by a claim is not sufficient to constitute disavowal. I also understand that even when statements during prosecution history do not rise to the level of unmistakable disavowal, they still inform the meaning of the claim. I further understand that arguments or amendments made concerning a particular patent application can be instructive as to the meaning of like terms in other related patent applications.

19. I understand that the claim language must, when viewed in light of the specification and prosecution history, inform those skilled in the art about the scope of the invention with reasonable certainty, otherwise the claim is invalid as "indefinite." I understand that this requirement assures that claims in a patent are sufficiently precise to permit a potential competitor to determine whether he or she is infringing. I understand that a claim is "indefinite" if its language might mean several different things and no informed and confident choice is available among the contending definitions, including based on information already well known in the art.

20. I also understand that terms of degree in claims, which call for a comparison against some baseline, are "indefinite" if they do not provide objective boundaries for those of skill in the art when read in light of the specification and prosecution history. I understand that absolute or mathematical precision is not required. I understand that examples in the specification may or may not allow a person of ordinary skill in the art to understand the

objective boundaries of a claim term, depending on whether the examples provide sufficient guidance as to the scope of the term. I also understand that, if the meaning of a claim term depends on the unpredictable vagaries of any one person's opinion, then that term is "indefinite."

V. LEVEL OF SKILL IN THE ART

21. My opinions herein are rendered with respect to the qualifications of a person having ordinary skill in the art ("POSA") to whom the Patents-in-Suit are directed. I understand from counsel that a POSA is a hypothetical person who possesses an ordinary level of skill and experience in the relevant art. As mentioned above, the relevant field for the Patents-in-Suit is DRM.

22. For purposes of claim construction, I have been asked by counsel to assume a relevant timeframe of June 16, 2011, the filing date of the '116 Patent. I take no position as to whether any of the Patents-in-Suit are entitled to their claimed priority date.

23. I understand that the following factors may be considered to determine the appropriate level of a POSA: (a) the types of problems encountered by those working in the field and prior art solutions thereto; (b) the sophistication of the technology in question, and the rapidity with which innovations occur in the field; (c) the educational level of active workers in the field; and (d) the educational level of the inventor.

24. After considering these factors in the context of the Patents-in-Suit, I conclude that a POSA in this time frame would have had a bachelor's degree in computer science or related field, and two years of experience working in the field of digital rights management. This description is approximate and additional educational experience could make up for less work experience and vice-versa.

VI. TECHNOLOGY BACKGROUND

25. To provide a framework for my opinions, the following is a brief explanation of technology concepts relevant to the claims of the Patent-in-Suit that a POSA would have been familiar with in the 2011 timeframe.

A. Digital Rights Management (DRM)

26. By 2011, the field of DRM had been well established for over a decade. As a field of study, DRM dates back to the mid-1990s, when the first important conferences were held and whitepapers on the subject were published.[1] The first commercial DRM systems were released in the late 1990s by companies including IBM, Xerox, Liquid Audio, and FileOpen Systems. Dozens of DRM technologies have been introduced since the late 1990s. Early DRM implementations with wide adoption include Apple's FairPlay DRM system, which launched in 2003 and was used by its iTunes system for purchase of digital music files for playback on iPods and computers. Microsoft was also an early player in DRM technology — it launched its Windows Rights Management Services DRM technology for Microsoft Office files and other documents in November 2003,[2] after releasing earlier DRM technology for audio, video, and e-book files.[3]

27. DRM techniques were invented to address the challenge of digital data—that it can be copied infinitely many times at virtually zero cost without any degradation in the quality of the copies. This was particularly seen as an issue for copyrighted works such as text, music,

[1] As an example of the former, the conference Technological Strategies for *Protecting Intellectual Property in the Networked Multimedia Environment* took place in Washington, DC, in late 1993. As an example of the latter, *Letting Loose the Light: Igniting Commerce in Electronic Publication*, by Dr. Mark Stefik of Xerox PARC research labs, was first published in 1994.
[2] https://news.microsoft.com/2003/11/04/microsoft-releases-windows-rights-management-services-for-windows-server-2003/
[3] https://news.microsoft.com/2001/06/13/microsoft-drm-technologies-establish-foundation-for-emerging-internet-music-video-and-ebooks-industries/

and video, because digital technology made copyrights on such content much easier to infringe than with analog media formats such as paper, recording tape, phonograph records, photographic film, or videotapes. DRM technologies are designed to allow digital content to be provided to users while still *limiting* how the content may be used and/or distributed to others. DRM systems allow ongoing rights and restrictions to be specified and enforced on the protected content (e.g., restricting access to specific users and/or devices, restricting access to some expiring time period, restricting user's ability to edit/print/copy the protected content). Some DRM technologies can be used with any type of content, while others are specialized for only certain types. DRM can apply to any type of digital content, including the above types, as well as office files (word processing documents, spreadsheets, presentations, etc.) in corporate environments, and even commercially distributed software.

28. DRM generally refers to protecting digital content files by means of encryption so that only users or devices with appropriate credentials can decrypt the files and access the content in them according to certain restrictions. Files containing encrypted content for distribution in DRM implementations are often known by those skilled in the art as *packaged* files, or more simply as *packages*, because they often also contain other information, as described below.

29. Encryption has, of course, been in existence for centuries and used to protect secret messages, such as military and intelligence information, so that only the intended recipients can understand them. In the realm of digital data, encryption can be used to protect stored or transmitted content so that it can only be accessed by particular recipients — namely, those in possession of a *decryption key*. First, the original content (known in the art as *plain text*) is encrypted, which generates new data known as *ciphertext* that is a complex mathematical

transformation that depends on the original content and an *encryption key*. The reverse process is *decryption*, which takes as input the cipher text and a decryption key and uses a mathematical computation to regenerate the original plaintext content. The ciphertext appears to be gibberish and so can be safely transmitted or stored without revealing the original content, because only those who possess the decryption key can decrypt the data and recover their original contents. The encryption and decryption keys are numerical values that are required inputs to the encryption and decryption algorithms. Successful encryption algorithms are those that are virtually impossible[4] to reverse without knowing the decryption key.

30. There are two general types of cryptographic algorithms that are used to encrypt and decrypt files of digital content: *symmetrical*, also known as *symmetric-key*, and *asymmetric-key*, also known as *public-key* or *public/private key*. Both types of algorithms date back to the 1970s. In a symmetrical algorithm, the same key is used to encrypt and decrypt content. The most prominent symmetric-key algorithm used in DRM systems in 2011 (as well as today) is the AES (Advanced Encryption Standard), published in 1998 and adopted as a U.S. government standard in 2001.[5] In an asymmetrical algorithm, different keys, called public and private keys, are used to encrypt and decrypt content. These keys are mathematically related, though it is virtually impossible to derive one key from the other. In asymmetrical algorithms, one encrypts content with a public key (a key that anyone can access), and the corresponding private key is required to decrypt the content. Asymmetric-key algorithms in wide use in DRM systems in 2011 (as well as today) include RSA (Rivest-Shamir-Adleman, named for its inventors and published in 1977) and ECC (Elliptic Curve Cryptography, first published in 1985). An

[4] There is no such thing as a 100% "hack-proof" encryption algorithm in practice; but today's algorithms in wide use, such as those mentioned here, are considered unbreakable for all practical purposes if implemented correctly.
[5] AES replaced the older Data Encryption Standard (DES), which dates back to 1976.

advantage of asymmetric-key encryption over the symmetric-key method is that it enables an encryption key to be published without compromising the security of the corresponding decryption key. This means that someone can publish a message that only a recipient can read without having to know the recipient's decryption key.

31. With both symmetric and asymmetric encryption, the decryption key must somehow be conveyed to the intended recipient of the content in a secure manner. As we will see, there are various ways of accomplishing this.

32. Encryption is an effective means of restricting access to content, yet if used by itself, its efficacy ends once the user possesses the key. If the recipient is simply provided with a decryption key to use freely, then, once decrypted, the content file can be used, copied, or manipulated in whatever ways the user desires, as long as the user has the key, thus defeating the purpose of the encryption in controlling usage of content in a DRM system. For this reason, it is generally not acceptable to merely send a decryption key along with ciphertext, because then an adversary can simply intercept the file, use the key, and retrieve the plaintext for any use.

33. DRM systems can address this by providing various types of information in the content package that an authorized user (and *only* an authorized user) can use to obtain the decryption key. For example, this could be an Internet address (such as a URL) and/or a key identifier that could be used to retrieve the key from a server. It could be some encrypted cyphertext that could be used to obtain the decryption key, but then that decryption operation would require its own key. Any of these could be included within the content package or sent separately.

34. In general, DRM systems are designed with the assumption that the intended recipients of packaged content are not to be trusted to use the content beyond the rights that the

originator of the content grants them. For usage restrictions to be effective, DRM systems must integrate decryption mechanisms with software programs that monitor and control the intended uses of content, such as word processors, spreadsheets, image viewers, and media players, so that those applications can be trusted to control the recipients' use of content over time. Such applications can be versions or extensions of general-purpose applications, or they can be dedicated software for using DRM-packaged content. Such applications will decrypt content only as necessary to carry out the operation that the user chooses, such as read, view, play, edit, print, copy to clipboard (for pasting later), etc., and not make the decrypted content available to the user for uncontrolled use.

35. Thus, distributors of content can use DRM to impose specific restrictions on use of the protected content that are enforced in an ongoing manner. In addition to specifying which restrictions to impose, they can also include limitations or extents on those operations, such as by time (e.g., play this video file for a month) or by number of operations (e.g., print this document up to 10 times; copy up to 300 characters to the clipboard). These restrictions can apply to users (via any device the user uses), devices, or combinations thereof. DRM systems use various means to identify users and devices for these purposes.

36. DRM systems often tie usage restrictions to specific users (customers of a content retail service, employees of a company) and/or devices (their PCs, tablets, smartphones, set-top boxes, etc.). To do this, they use various schemes to identify users or devices to which restrictions apply, such as user or device identifiers. So, if multiple users get the same packaged file, only the one with the correct identifier (or the one whose device has the correct identifier) can use the content.

37. To specify the usage restrictions, DRM systems enable distributors of content to send user devices information on the rights being granted to them. These are typically contained in small files that contain descriptions of the rights being granted, along with decryption keys and other information (such as user or device identifiers). These small files are often called *licenses*. Other terms are used in particular DRM implementations (permits, tickets, etc.), but those skilled in the art understand "license" as a generic term for this.

38. Here is a description of how a typical DRM system works with the technology components described above. These components and techniques were all well-known and in common usage by the time the Patents-in-Suit were filed.

39. First, a user obtains a packaged file and wants to access the content it contains in a software application (a music player, word processor, image viewer, etc.) that has DRM capabilities built in.

40. The application looks for a license that grants rights to the files for the user and/or the device. The license may be stored within the packaged file or sent separately from the packaged file; or it may be already stored somewhere on the user's device.

41. If the application can't find an appropriate license, it must obtain one. The application will send a request for a license from a server operated by (or on behalf of) the content publisher, often known as a *license server*. The license request will generally include some information for the server to identify the appropriate license. The license request will generally also include some information about the user and/or device so that the license server can evaluate whether the license request should be granted.

42. The license server receives the request and checks to see whether the user and/or device is authorized to run the requested operation. If so, the server will retrieve or prepare a

license. Because a license file typically includes the content decryption key, the server itself needs a way to obtain that key. In some DRM technologies, the decryption keys are stored on the server (e.g., in a database), such that the server can look up and retrieve the appropriate key based on some information included in the request (e.g., an identifier). In other DRM technologies, the content decryption keys are not stored but rather generated at the server when needed. Other ways of generating content decryption keys were known in the art in the relevant timeframe. In any case, once the server has determined that the user/device is authorized to receive a license and has determined the appropriate license and decryption key, the server sends the license to the user's device, which may store it in an appropriate place on the device for later reuse.

43. The client DRM software will open and examine the license to determine whether the license gives the user permission to run the selected operation. If not, the operation is disallowed.

44. If the permission is granted, the DRM software will extract the decryption key from the license. Various ways to do this are known in the art, depending on the type of security that the DRM scheme gives to licenses and decryption keys. The software then uses the decryption key to decrypt the content for the sole purpose of running the user's selected operation. For example, if the content is text and the operation is "read," then it will decrypt the content, pass the result to the program's display function, and not store the decrypted content any further. If the operation is "print," then it will pass the decrypted content to a print driver (printer interface) and not store it any further.

45. I have reviewed Plaintiff PASafeShare LLC's Opening Claim Construction Brief ("Plaintiff's Opening Brief"). Plaintiff's Opening Brief contains a "Technology overview"

section at pp. 1-4. This explanation differs from the one offered here in various ways, and as I will explain, I disagree with certain aspects of it.

46. As a general matter, the examples given in the Plaintiff's technology overview are unnecessarily restricted to Microsoft Office applications such as Word, Excel, and PowerPoint. As mentioned above, many DRM technologies are designed to apply to a variety of types of content, not just "office documents" such as word-processing documents, spreadsheets, and presentations.

47. Beyond that, I find a number of inaccuracies in Plaintiff's technology overview, including but not limited to the following:

48. "Once encrypted, only users on the authorized user list can open or view the Word document." (Plaintiff's Opening Brief, p. 3.) This misstates the types of restrictions that a DRM system can place on operations on a content file. For example, restrictions on a word processing document (as well as other content types) can include printing and copying to clipboard. Such restrictions are not necessarily yes/no, on/off restrictions but can be limitations, such as print up to N times or copy up to N characters to the clipboard, as described above.

> "… [T]he content publisher's device may send the Word document as well as … an authorized user list. … [T]he content consumer's device generates a message including, among other things, … the encrypted authorized user list. … [T]he licensing server decrypts the authorized user list and compares the device's/user's identity to the authorized user list. If the device's/user's identity matches an authorized user on the list, … the license server sends a use license to the content consumer, giving the content consumer permission/authorization to use the decryption key and open the document."
>
> (Plaintiff's Opening Brief, pp. 3-4.)

49. This is not a typical protocol in a DRM system. Among other things, "authorized user list[s]" are often too large to send back and forth between clients and servers in this way; and exposing authorization information to one user about other users is considered a security

hole. A more typical protocol involves the server maintaining its own list or database of rights authorized for users/devices on content files, as described above, checking authorizations locally on the server, and not sending this information out over the network.

B. Software Instructions vs. Data

50. The design of computers generally includes processors, which execute *software programs* (consisting of instructions), which in turn use or operate on *data*. Computer processors are designed to execute software instructions, which in turn are the building blocks of software (i.e., a program). For example, the Microsoft Computer Dictionary, 5th edition, 2002 ("Microsoft Computer Dictionary") defines "software" as "Computer programs; instructions that make hardware work." (Microsoft Computer Dictionary, p. 489); it defines "program" as "A sequence of instructions that can be executed by a computer. . . *Also called*: software" (Microsoft Computer Dictionary, p. 424); and it defines "execute" as "To perform an instruction. In programming, execution implies loading the machine code of the program into memory and then performing the instructions." (*Id*., p. 200). For example, applications such as Microsoft Word, email software such as Microsoft Outlook, web browser software such as Microsoft Internet Explorer, and the calculator app built into Microsoft Windows are all executable programs.

51. Other collections of information stored on a computer are simply data, which can be used or operated on by programs, but isn't itself executable. Data can exist as files, such as a digital image in JPEG format that requires an image viewer program such as Microsoft Photos or Adobe Photoshop, or HTML files that can be opened in a web browser such as Microsoft Edge or Google Chrome. Microsoft Computer Dictionary defines "data file" as "A file consisting of data in the form of text, numbers, graphics, or structured combinations thereof, as distinct from a program file of commands and instructions. *Compare* program file." (Microsoft Computer Dictionary, p. 142, italics in original.) In other words, data files are not executable software but

contain data that can be used by executable programs. For example, data such as email messages and associated data (e.g., sender and recipient addresses, subjects, date sent, date received, and importance flags) can be sent, received, or displayed within Microsoft Outlook; or data can be user input, such as numbers typed into a software calculator application. A program takes actions and can use the data in those actions: for example, an email program receives, sends, displays, stores, and sorts email messages; it can decide how or if to display the displayable data (e.g., how to designate "high importance" messages; whether to sort messages by date or by sender). In all these cases, the actions are performed when the program's instructions are executed by the computer.

52. In other words, a POSA would understand that there is a fundamental difference between programs and data files — e.g., programs run on the computer (or equivalent device), i.e., contain executable software instructions that are executed by the computer's processor, while data files are simply *used* by executable programs.

C. Software Instructions and DRM

53. Some DRM system designs incorporate content packages that include their own executable software for obtaining licenses and, in some cases, controlling access to the protected content. These DRM implementations do not rely on external software on the user's device for such functions. This was a fairly common design choice, for example, in DRM systems for commercial software distribution.

54. Other designs rely on such external software, which must be separately installed on the user's device before the DRM can be used. Two examples of this technique, Ginter and Wajs, existed well before 2011 and will be discussed below. In systems like these, the content packages are data files and do not contain executable software, instead relying on pre-existing separate software.

55. The tradeoffs between the two techniques were understood. For example, DRM systems with self-contained license requesting software could always be certain that the license requesting software was delivered directly from the content distributor and was compatible with the server, was not out of date, and was less likely to have been hacked. On the other hand, if the DRM system was intended to work across multiple client operating platforms (e.g., on Windows PCs, Macs, Android phones, iPhones, etc.), they required that the software distributed in the content package was the correct version for the user's device's operating platform. DRM system designs that incorporated non-executable content packages enabled the packages to be smaller, taking up less network traffic, and enabled different versions of external DRM client software to be distributed that worked on different operating platforms.

D. Virtual Private Networks (VPN) and Secure Tunnels

56. When one computer communicates with another over a network, the communication may be subject to eavesdropping by potentially malicious third parties. Therefore, there are various techniques known in the art to secure digital communications sent over networks. The term "tunneling" in computer networking generally means to encapsulate one communications protocol within another. The Microsoft Computer Dictionary defines "tunneling" as "A method of transmission over internetworks based on differing protocols. In tunneling, a packet based on one protocol is wrapped, or encapsulated, in a second packet based on whatever differing protocol is needed in order for it to travel over an intermediary network. In effect, the second wrapper 'insulates' the original packet and creates the illusion of a tunnel through which the wrapped packet travels across the intermediary network." (Microsoft Computer Dictionary, p. 532.) As a simple example, assume that there is a communication protocol P1 that has a MessageBody field. Someone wants to send a message in a protocol P2 by tunneling it through P1. To do so, they could construct a message in P1 with their P2 message as

the MessageBody field in P1 and send it. When the message is received, the recipient decodes the message using protocol P1, retrieves the MessageBody field from the message, and gets the message in protocol P2.

57. "Secure tunneling" is a generic term for using tunneling to protect a message from eavesdropping, typically by using encryption. Secure tunnel is sometimes used as a synonym for a virtual private network (VPN) or to denote a superset of VPN functionality. A VPN is a secure tunnel that is usually used specifically in the Internet context to preserve privacy. It is a network communication scheme that uses encryption to foil eavesdroppers, thereby emulating a private network where eavesdropping is physically impossible. The Microsoft Computer Dictionary defines "virtual private network" as, in relevant part, "Nodes on a public network such as the Internet that communicate among themselves using encryption technology so that their messages are as safe from being intercepted and understood by unauthorized users as if the nodes were connected by private lines." (Microsoft Computer Dictionary, p. 554.)

58. A POSA as defined above generally understands that if a user uses secure tunneling or a VPN means to connect to a server or other network resource, then the user's connection is more secure than if secure tunneling or a VPN were not used.

VII. OPINIONS ON CLAIM TERMS

A. **"software instructions . . . executed by a processor"** (all asserted claims)

59. I agree with Microsoft that this term should be given its plain and ordinary meaning, which a POSA would understand to be "executable software instructions".

60. Clarifying that the plain and ordinary meaning is "executable software instructions" is important to assist the trier of fact in making the distinction that a POSA would make between programs that contain executable software instructions and data files that do not, as explained at ¶¶ 38–39 above. It is also important because the intrinsic evidence shows that

POSA repeatedly made this very same distinction in the file history and relied on the same distinction to try to distinguish prior art.

61. That the patentee intended to limit his invention to protected document packages (PDPs) that contain executable software instructions is evident from both the plain language of the claim and the prosecution history of the ’116 Patent (from which the other Patents-in-Suit were continuations-in-part). PDP is Plaintiff’s terminological equivalent of packages as described at ¶ 15 above. The claims explicitly require the PDP to include “software instructions which, when executed by a processor” cause the device to generate a license request. A POSA would understand that the plain and ordinary meaning of the limitation is that the PDP must include software (i.e., a program) whose instructions are executed by the processor to generate a license request. In other words, the executable software instructions for generating a license request must be included within the PDP itself. This is supported in the specification, which explicitly describes embodiments that use “an executable PDP or an executable container program including the PDP information.” ’116 patent at 7:62–63.

62. During the course of the prosecution, Plaintiff interpreted this claim limitation exactly in line with this plain and ordinary meaning and explicitly disclaimed functionality where a non-executable PDP relied on software instructions that were separate from the PDP (e.g., preinstalled client software). The prosecution history shows that Plaintiff repeatedly distinguished its purported invention from prior art that relied on external software to decrypt encrypted content packages and control access to content rather than on executable software contained in the packages themselves.

63. One such piece of prior art was published as U.S. patent application 2002/0048369 to Karl Ginter et al (“Ginter”). After the ’116 patent claims had been amended to

require "said PDP including software instructions which, when executed by a processor" generate a CCLR [Content Consumer License Request], the Examiner rejected the claim over Ginter. In an interview, to try to overcome the rejection, Applicant argued:

> [T]he VDE black box container of Ginter was only functional in a VDE environment after a preinstalled software package was function [sic] on each subscribed users computer *which is different from the claimed PDP which requires no preinstalled software prior to the receipt of the PDP . . . .*"

'116 Patent File History, Interview Summary (May 6, 2014) (emphasis added)."

The Applicant later stated in a response:

> "… Ginter provides an environment within which widely deployed VDE [Virtual Distribution Environment] client applications provide a mechanism for subsequent secure distribution of content thereby. By contrast, the claimed invention contemplates providing encrypted content via a PDP on an as-needed basis, wherein *there is no need for a pre-installed client mechanism as provided in Ginter.*
>
> In contrast to Ginter, the claimed invention does not require the use of a preinstalled client application such as contemplated by Ginter. The VDE system of Ginter contemplates a secure environment in which a client-side application is widely distributed and installed (preferably preinstalled) on client devices prior to such client devices receiving content in a secure manner. …
>
> In contrast to Ginter, the claimed invention provides within the PDP file the encrypted content which may be consumed by an authorized user, as well as a mechanism by which a proposed authorized user may become an authorized user capable of consuming the content. Ginter provides no such mechanism since such a mechanism is unnecessary due to the reinstallation of VDE client applications which pre-authenticate users for particular types of content."
>
> In contrast to Ginter, authorization to view content within a PDP attaches to the user the virtual possession of a Content Consumer License (CCL). *No separate client application* such as the VDE client application is necessary for such authorization.
>
> In contrast [to] Ginter, the claimed invention provides <u>within the content-bearing PDP file itself</u> a mechanism by which a proposed authorized user becomes an authorized user. Ginter provides for user authorization via the VDE client application, which authorization occurs <u>prior to receiving</u> the securely-delivered content.

'116 File History, Office Action Response at 12 (May 15, 2014) (underlining in original). He also said that in "the claimed embodiments . . . strict control is provided via the PDP including both encrypted content and a mechanism providing a recipient with the ability to 'generate a Content Consumer License Request (CCLR) identifying said PK' . . . ." *Id.* at 13 (underlining in original). The applicant was quite clear that "the claimed invention does not require the use of a preinstalled client application" and "provides within the content-bearing PDP file itself a mechanism by which a proposed authorized user becomes an authorized user." '116 File History, Office Action Response at 12 (June 16, 2011) (underlining in original).

64. Another relevant prior art reference in the 116 File History was published U.S. patent application 2006/0080259 to Andrew Wajs ("Wajs").[6] The Examiner rejected most of the Applicant's claims over Wajs ('116 File History, pp. 413-422.) In his response to the rejection, the Applicant distinguishes his invention over Wajs by stating:

> "Unlike the claimed invention, Wajs contemplates a *relatively standard 'trusted agent'* type of digital rights management (DRM) scheme for use in a cellular telephone system in which encrypted content may be decrypted by clients within the cellular telephone system (i.e., handsets or mobile devices of the cellular telephone system users). … The extent of such consumption or use of content by a client is defined by a rights object cryptographically bound to a target DRM agent and including an agent-specific decryption key for that content."

'116 File History, Office Action Response at 10 (Dec. 14, 2014)

> **Wajs fails to disclose or suggest at least the limitations of "said PDP including software instructions which, when executed by a processor at a proposed authorized user device, cause said proposed authorized user device to generate a Content Consumer License Request (CCLR) identifying said PK;**
> **. . . .**
>
> In contrast to the above-quoted claim language, the Wajs secure content package does not include any such software instructions . . . .

[6] Wajs was the Chief Technology Officer of Irdeto, a Dutch company that provided (and still provides) DRM technology for pay television.

> In particular, even if the rights data object could be construed as even remotely similar to the CCL (it is not), the mechanism by which a rights data object is obtained by Wajs is via agent function software code stored in the EEPROM 40 of a handset (see, e.g., paragraph 56 of Wajs)

Id. at 12–13.

> It is noted that the claimed invention contemplates distribution of content without the need for an embedded agent program or similar *pre-existing* software at the client device. (e.g., a handset).

Id. at 12.

65. In other words, the Applicant again distinguishes his purported invention from prior art on the grounds that the alleged PDP did not have executable software instruction for generating a license request *within the PDP* and instead relied on pre-existing software at the client device (which the "claimed invention" does not require).

66. Applicant repeatedly took this same position over the course of the prosecution history, clearly reading the "software instructions . . . executed by a processor" to require executable software instructions (i.e., *a program*) within the PDP and expressly used that position to draw distinctions against the prior art. For example:

> In contrast to the above-quoted claim language, Pravetz does not contemplate the situation where a user not yet authorized to view/present content ***executes a program*** associated with that content (i.e., ***in the same package***) ***such that a CCLR is generated*** to retrieve thereby a CCL enabling the viewing/presenting of that content by the user."

'116 Patent File History, Office Action Response at 10–11 (Oct. 11, 2013) (emphasis added).

> The claimed PDP itself *includes* encrypted content *and* PK for decrypting/presenting the content, *and* computer instructions for generating a CCLR at a client device (emphasis in original).

'116 Patent File History, Office Action Response at 11 (March 16, 2015) (emphasis in original)

> "[T]he claimed invention requires a protected document package (PDP) [that] contains: 1. encrypted content. 2. a Publisher key (PK) to decrypt said content. 3. code to generate Content Consumer License Request (CCLR)."

'116 Patent File History, Interview Summary (Feb. 23, 2016).

67. In light of the explicit claim language and repeated statements through the '116 patent's prosecution history, a POSA would understand the claim language to require exactly what it says — i.e., that PDP must include executable software instructions for generating a license request.

68. Plaintiff's suggestion that "software instructions are simply instructions that cause a program to perform different types of task(s)" is both vague and incorrect. For starters, it reflects a fundamental misunderstanding about the difference between *software* and *data*. As described above, "software" and "software instructions … executed by a processor" refer to computer *programs*. For example, the Microsoft Computer Dictionary, 5th edition, 2002 ("Microsoft Computer Dictionary") defines "software" as "Computer programs; instructions that make hardware work.") (Microsoft Computer Dictionary, p. 489); defines "program" as "A sequence of instructions that can be executed by a computer. . . Also called: software" (Id., p. 424) and defines "execute" as "To perform an instruction. In programming, execution implies loading the machine code of the program into memory and then performing the instructions." (Id., p. 200). Nothing in the specification or file history suggests that the term "software instructions . . . executed by a processor" would be interpreted as anything else. To the contrary, the file history makes clear that this was exactly how the terms were meant to be interpreted.

69. In contrast, Plaintiff admits that it is now trying to read the claim to cover non-executable data. For example, Plaintiff argues that "an importance flag in an e-mail" could somehow be considered a "software instruction" that is "indirectly executed by a computer's CPU" (Plaintiff's Opening Brief at 8), *even though it is not part of any software*. An e-mail message is not "software" and does not contain "software instructions" — it is just data. The

different parts of an e-mail message are just different data fields (e.g., subject, to, from, bcc, date received, importance, body, etc.). One needs *software* to actually do anything with that data (i.e., view, send, write an e-mail), and that software is what provides the *software instructions* — i.e., the executable program instructions that determine what the program will actually *do* with the data and how it will do it. For example, an email program can do whatever it wants with the e-mail data that it receives — that is up to the software. An "importance" header field in an email is not a "software instruction" — it is simply many non-executable data fields that make up the e-mail, and different email programs can choose to do various things with that information (e.g., display a star or an exclamation point, make an alarm sound, or do *nothing at all*). Of course, the results of a program's software instruction will depend in part on the data that it is manipulating, but that doesn't somehow make the data field a "software instructions." The software instructions for Adobe Photoshop don't change just because you open a different image file — the only thing that has changed is the data provided to the software. Similarly, e-mail program's software instructions do not change every time a new e-mail is received.

70. Plaintiff's suggested interpretation of the "plain and ordinary meaning" of "software instruction … executed by a processor" is completely divorced from the facts and the Applicant's numerous disclaimers in the file history. It is also extremely vague — if something that was not a part of any software could somehow still be considered a "software instruction," it is unclear what would or would not be covered.

B. "Publisher Key (PK)" (all asserted claims)

71. Plaintiff proposes to construe this term as "Information for decrypting content from a content publisher and containing usage restriction(s) on the content[.]" Microsoft finds this term to be indefinite.

72. I agree with Microsoft that the term is indefinite and that Plaintiff's proposed construction is improper. My understanding of the law, as provided by counsel, is that in defining a term in a patent claim, one must first look to the intrinsic evidence, namely the patent specification and then the prosecution history. I have reviewed the patent specification and the file history, and while I have found various (often inconsistent) references to the coined term "Publisher Key," I have not found any disclosures that define the term or would otherwise provide a POSA with reasonable certainty about what the term would cover.

73. I am also informed that if the intrinsic evidence does not define a claim term, then one may look to the extrinsic evidence for a definition. Yet "publisher key" has no specific meaning in the art; it is not even analogous to a generic, non-implementation-specific term such as "license" (see ¶ 24 above) or "package" (see ¶ 15 above) that is known in the DRM field.

74. As described above, "decryption keys" were certainly known in the art, but the specification never explains the relationship, if any, between a "Publisher Key" and the content decryption key. One might think that the Publisher Key was simply another name for the decryption key, but the specification describes the Publisher Key being used in a manner that a POSA would understand would not make sense if it were the decryption key itself. For example, the specification discloses comparing reading the PK from a newly received PDP and comparing it to the PK in any existing stored license files to see if they match, and if they do not, requesting a license:

> "If the specified publisher key of the protected document package does not match the Publisher Key of any of the Publisher License Files on the user's computer, or if the date or license type is not valid, then the end user will be instructed to create a license request and send it to the content owner."

('116 Patent at 11:40-45 and similarly at 11:40-45.)

The device must be able to read the PK in the PDP if it expected to compare it with another value (i.e., to a PK in a license file). Yet, at the point this comparison is made, the device reading the "Publisher Key" from the PDP has not yet been authorized to use the packaged content. We know this because the comparison is described as how to *determine* if the device already has a license (i.e., has already been authorized). That might be acceptable if the Publisher Key were just some kind of identifier, but it would be extremely odd if the Publisher Key were the actual decryption key that enabled decrypting the content. If so, that would mean that the decryption key would be stored in the PDP alongside the content it was encrypted in a manner where *any* device that receives the PDP would be able to freely read the decryption key. But that would then mean that any device that receives the PDP could use the decryption key to decrypt the protected content, whether or not it had a matching license. That would render the encryption rather pointless (since encryption only works if the decryption key is kept secret) and would present a glaring security hole in the DRM system. So this description, among others, casts serious doubt about what the Publisher Key is supposed to be and how a POSA would be expected to identify a "Publisher Key" from anything else.

75. Similarly, meaning of Publisher Key is made even more ambiguous by the limitations in '116 Patent dependent claims 3 and 4, which require that the PK "define" an "authorized use" ("one or more of an authorized presentation start time, an authorized presentation start date, an authorized presentation expiration time, and an authorized presentation expiration date" (Claim 3); "one or more of a password protection layer, an encryption type, a presentation program type, authorized user device geographical location, and an authorized user device IP address range." (Claim 4.)) A content decryption key is just that—a key (see ¶ 16 above)—so it cannot contain such other information. The specification is otherwise silent about

what this Publisher Key is and does not contain any descriptions of the Publisher Key "defining" authorized uses.

76. Because "Publisher Key" is not a term of art, is never defined, and is used inconsistently, it is my opinion that a POSA would be unable to ascertain the scope of the claim with reasonable certainty. Put simply, a POSA would have no way of evaluating, for any given system, whether or not or it includes a "PK" within the meaning of the claims.

77. In addition, I find Plaintiff's proposed construction to be vague and ambiguous. It is not clear what would or would not be covered under this proposed construction. On the one hand, "[i]nformation for decrypting content … and containing usage restriction(s) on the content" suggests a license, which a POSA would understand contains a content decryption key (or a key identifier, or an encrypted key) along with usage rights information, as explained above. However, the specification already discloses a Content Consumer License (CCL), which is also used to restrict access to content. Figure 6 in the '116 Patent shows that a CCL contains a PK (which, as discussed above, is not an encryption key) and, separately, what a POSA understands to be license terms: start and expiry dates, and an indication of whether the license pertains to the user, the device, both, or neither. Otherwise, Plaintiff's construction offers no indication of what the PK actually is or contains and just raises more questions about what would count as "information for decrypting content," what "from a content publisher" actually requires, and what it means for the PK to "contain[] usage restrictions on the content."

C. "enabling an authorized user communicating via a secure tunnel or virtual private network to modify PDP content via an editing mode associated with said LCV" ('961 patent, claims 6, 14; '848 patent, claims 15, 18)

78. Plaintiff proposes to construe this term by its "plain and ordinary meaning." Microsoft proposes to construe this term to mean "if an authorized user is communicating via a

secure tunnel or virtual private network, then enabling that authorized user to modify PDP content via an editing mode of the LCV."

79. I agree with Microsoft's proposed construction. A POSA would understand the claim language to specify a clear causality between communicating via a secure tunnel or VPN and obtaining rights to edit content. The claim language presents this phrase as a single claim limitation with related components: "an editing mode associated with said LCV" is being enabled for a specific user — namely, "an authorized used communication via a secure tunnel or virtual private network." This same relationship between the editing mode of the LCV and an authorized user connected via secure tunnel or VPN is described in the specification of the '961 Patent: "an authorized user receiving a PDP via a secure tunnel or VPN is enabled to modify the PDP content via interaction with a server or content provider such that subsequent distribution of the PDP will include such modifications." ('961 Patent at 15:45-49.)

80. A POSA understands that conferring editing rights on a user generally requires more trust in that user than, say, viewing or reading rights. It would therefore make sense to a POSA to condition editing rights on the security of users' connections. As explained at ¶ 58 above, one purpose of a VPN is to emulate the security level of a physical connection to a server over a network such as the Internet.

81. Plaintiff argues against this construction on the basis that it limits the scope of the claim to "only those users who communicate via a secure tunnel or VPN." (Plaintiff's Opening Brief, p. 9.) I disagree; there is nothing in Microsoft's construction limiting editing rights to *only* those users who communicate via a secure tunnel or VPN. On the contrary, Microsoft's construction allows for the possibility that other users (not connected via VPN) can obtain editing rights. A POSA would understand that an implementation of the claims in the Patents-in-

Suit may, for example, grant editing rights to users who are connected to a server either via a physical connection *or* via a VPN/secure tunnel.

82. Finally, a POSA would not understand the term in the manner that Plaintiff proposes — i.e., that "*Any* authorized user, *regardless of how he/she communicates with a network*, can edit content (provided the editing function has not been restricted)." (Plaintiff Opening Brief at 10). This makes no sense in light of the claim language, which states that the editing mode is enabled for "an authorized user communicating via a secure tunnel or virtual private network."

D. "enhanced user authentication or authorization (EUAA) information" ('961 patent, claims 2-4, 12, 13)

83. Plaintiff proposes to construe this term as "Information about a user that identifies the user for authentication or authorization." Microsoft disagrees with Plaintiff's construction and finds this term to be indefinite.

84. I agree with Microsoft that the term is indefinite and that Plaintiff's proposed construction is improper. "Enhanced" is a term of degree and of comparison; something can only be "enhanced" with respect to something else (i.e., to some baseline). The specification does not explain what this baseline is or what standard a POSA would use for deciding whether any given "user authentication or authorization information" in a DRM system would be sufficiently "enhanced" to be covered by the claims.

85. As an initial matter, a POSA would understand that there is a difference between "authentication" (proving that someone is who they say they are) and "authorization" (being granted a right to do or obtain something). The claim term blends the distinct concepts without providing any explanation.

86. At one point, the specification enumerates types of information that could be used as "EUAA information":

> "EUAA information may include content consumer details including personal identification indicia in addition to computer, domain, hardware, and software identification, including, but not limited to biometric information, fingerprints, face and eye recognition, handwritten signature and its properties (gait and pressure in signature), externally provided security encryption keys (e.g., Fortezza crypto cards), Voice recognition, or other personal identification methods."

('961 Patent at 14:65-15:6)

87. Yet the foregoing conflicts with other disclosures in the specification, resulting in ambiguity and confusion. For example, the specification also discloses that "the [Content Consumer License Request] includes … content consumer details such as identification of proposed authorized user(s)," which is not described as EUAA information ('961 Patent at 9:21-24 and 9:60-64). A POSA understands that information such as "biometric information, fingerprints, face and eye recognition, [and] handwritten signature and its properties (gait and pressure in signature)" are examples of "identification of proposed authorized user(s)" and that the latter term is more general than the more specific "personal identification indicia" in the above; therefore, the boundary between "basic" and "enhanced" user authentication or authorization information is unclear.

88. As another example, the specification discloses:

> "… the security/verification mechanism may comprise *any of* an Enhanced User Authorization and Authentication (EUAA) mechanism, a biometric mechanism, a smart card or universal serial bus (USB) security mechanism, recipient GPS coordinates *or* other security/verification mechanism."

('961 Patent at 17:59-64 and 18:29-35, emphasis added.)

89. In this case, the practitioner is advised to choose *between* a security/verification mechanism that comprises an EUAA mechanism *or* a biometric mechanism, even though the specification instructs, as shown above, that EUAA information may *include* biometric information.[7]

90. Plaintiff's proposed construction purports to eliminate the confusion and ambiguity by pretending that "enhanced user information" means "information about a user," without any qualification. This ignores any notion of a baseline from which "enhanced" would be distinguished, thus rendering the term "enhanced" meaningless. A POSA would understand the term "enhanced" in the claim language to be there for a reason and so would not understand *any* information about a user to constitute "*enhanced* user authentication or authorization information."

91. Finally, I have reviewed the prosecution history of the '961 Patent ("'961 File History") and found nothing in it that would assist a POSA in clarifying the meaning of this term beyond what is in the specification.

VIII. CONCLUSION

92. In my opinion, the claim phrase "software instructions . . . executed by a processor" should be construed according to its plain and ordinary meaning, which is "executable software instructions."

93. In my opinion, the claim phrase "enabling an authorized user communicating via a secure tunnel or virtual private network to modify PDP content via an editing mode associated with said LCV" should be construed to mean "if an authorized user is communicating via a

[7] A POSA would also understand that "fingerprints, face and eye recognition, handwritten signature and its properties" are all *examples of* biometric information.

secure tunnel or virtual private network, then enabling that authorized user to modify PDP content via an editing mode of the LCV."

94. In my opinion, the claim terms "Publisher Key (PK)" and "enhanced user authentication or authorization (EUAA) information" are indefinite.

I declare under the penalty of perjury under the laws of the United States that all statements made in this Declaration are true and correct.

Executed on December 22, 2020 in New York, NY.

William Rosenblatt